



HAL
open science

Sector size in Magnetic, Electronic and Optical media

Charbel Tannous

► **To cite this version:**

Charbel Tannous. Sector size in Magnetic, Electronic and Optical media. Maitrise. Technologie des Médias, Brest, France. 2018, pp.16. <hal-03882390v2>

HAL Id: hal-03882390

<https://cnrs.hal.science/hal-03882390v2>

Submitted on 18 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Sector size in Magnetic, Electronic and Optical media

C. Tannous 

Université de Brest, Lab-STICC, CNRS-UMR 6285, F-29200 Brest, FRANCE

(Dated: [October 18, 2024](#))

Information storage is reviewed from the magnetic, electronic and optical points of view. Sector size being the unit of information has a paramount importance in storage and retrieval of information. In this work we concentrate on sector coding and its length optimization in order to obtain the lowest BER (Bit Error Rate) during reading and writing of Information in magnetic, electronic and optical digital media.

PACS numbers: 75.70.-i, 84.40.Ua, 42.79.Vb, 42.70.Ln, 73.63.-b, 81.07.-b

Keywords: Magnetic Materials, Telecommunications, Optical storage devices, Optical media, Nanoscale materials electronic transport, Nanoscale materials fabrication and characterization

Contents

I. Introduction	1
II. Line Coding	2
III. Error Correction and Coding	2
IV. Magnetic Storage	4
A. Historical CHS address coding and sector size	4
B. Error Correction codes for magnetic recording	5
C. Sector size increase issues	7
V. Electronic Storage	8
A. Error Correction codes for NAND Flash	9
VI. Optical Storage	11
A. Error Correction codes for Optical storage	11
A. Detailed description of Line and RLL codes	11
B. Galois, CRC and Partial response polynomials	12
C. Reed-Solomon codes	15
References	15

I. INTRODUCTION

For a long time, the unit of binary information (sector) in Hard Disk Drives (HDD) has been fixed at 512 Bytes (about $\frac{1}{2}$ k-Byte or roughly 4k-bits) of data whereas in CD/DVD/Blu-Ray media a sector has been taken at about 2048 Bytes (2k-Bytes). In Flash electronic memories, a sector might be of 128, 256, 512...4096 Bytes. In this work we are interested in sector size meaning and optimization from the coding as well as error detection and correction points of view in magnetic, electronic and optical storage media.

Binary information storage is based on many steps of coding and equalization procedures. Coding encompasses signal and transmission channel in terms of binary representation, adaptation (line or modulation), error correction and pre-distortion (partial response). Another important operation is data compression that we do not tackle in this work.

Processing of binary information in terms of writing and reading for storage and transmission is briefly displayed in fig. 1.

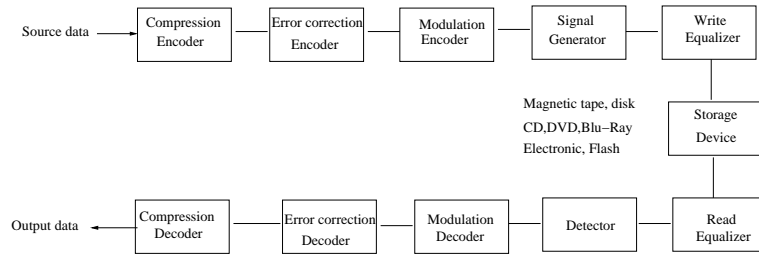


FIG. 1: Coding chain in magnetic, optical and electronic recording.

It is important to consider and evaluate sector size since it impacts coding efficiency gain, increased ECC (Error Correction Coding) performance and better handling of burst errors.

II. LINE CODING

The first step in handling binary information for storage and transmission is line coding. Line coding (or Modulation coding) is applied in order to achieve at least three goals:

1. Automatic (self-contained) clock recovery in order to avoid an additional signal carrying time coding for the signal. This is detected when there is a peak in the spectrum.
2. AC coupling of bit signal: Since a raw bit is an antisymmetrical signal, harmonic distortion of the signal might occur. Symmetrizing the signal by removing the DC component makes it closer to a purely harmonic signal making it less prone to harmonic distortions.
3. Placing several zeroes between consecutive ones in order to avoid overlap and interference between consecutive ones, called ISI (Inter-Symbol Interference).

Note: In magnetic recording, if we decrease the number of $0 \rightarrow 1$ transitions, we might end up reducing errors while ignoring AC coupling.

III. ERROR CORRECTION AND CODING

The description of digital error management starts with a statistical assessment of error rates observed in various communication channels [2], networks or storage media as given in Table I.

Media or Communication Channel	Error Rate
HDD	$> 10^{-4} \sim 10^6$ errors in 10 GB
DAT (Digital Audio Tape)	10^{-5} or 2 errors per second at 48 kHz
CDROM	10^{-5} or 7kB errors on a 700 MB CD-ROM
Blu-ray disk	$< 2 \times 10^{-4} \sim 1.6$ MB errors on a 32 GB disk
Flash electronics	$\approx 10^{-5}$
Solid-State Drives (SSD)	$< 10^{-5}$
Semiconductor memories	$< 10^{-9}$
Infrared remote control	10^{-12}
Optical fiber communications	10^{-9}
ADSL	10^{-3} to 10^{-9}

TABLE I: General BER in different media (see Mueller [3]).

In order to detect and correct the errors, we need EDC and ECC (Error detection and correction codes) algorithms classified as displayed in fig. 3.

We describe first magnetic storage, then electronic and finally optical.

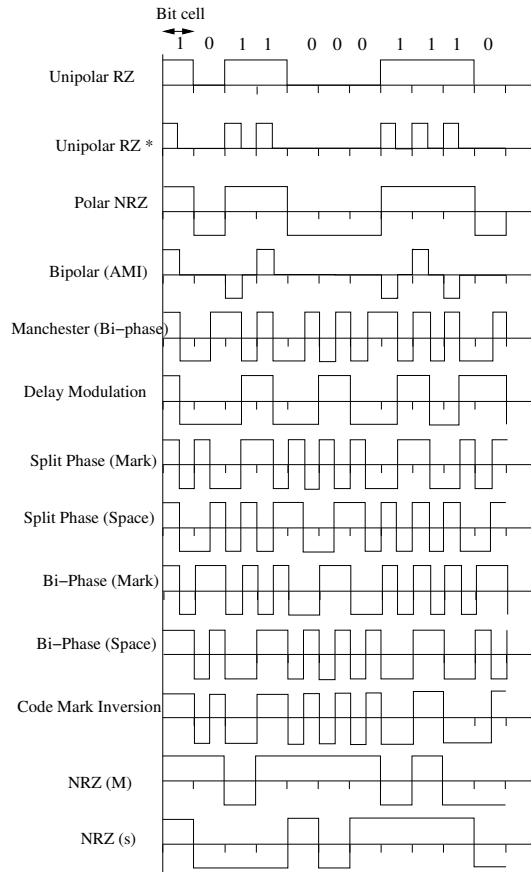


FIG. 2: Modulation coding of information with associated time dependent waveforms. M and S are Mark and Space the older notation for 1 and 0. T_b is bit period. When a coding goes from M to S the roles of 1 and 0 are interchanged. (*) is a unipolar RZ waveform in which 1 is represented by a non-zero voltage level at half the bit period. Adapted from Ref. [1].

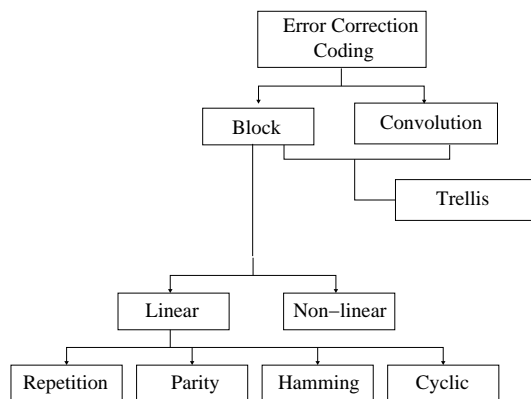


FIG. 3: Classification of ECC codes. Cyclic block codes comprise Reed-Solomon and BCH (Bose-Chaudhuri-Hocquenghem) codes. Block codes are characterized by (n, k) where n is the number of data bits and k the number of parity bits with no memory. Convolutional codes are characterized by (n, k, m) with n the number of data bits and k the number of parity bits and m the number of memory steps. Note that a BCH $(n, k; t)$ code uses Galois polynomials (see Appendix) $GF(2^m)$ with n bit codeword length and k bit message length can correct up to t error bits. Trellis codes are a mixture of Block and Convolutional.

IV. MAGNETIC STORAGE

Magnetic line coding displayed in fig.4 is concerned with flux change triggered by magnetic media moving with respect to the read head device.

Presently an HDD sector can be as large as 4096 Bytes (4k-Bytes) and the question is why it took so long and what are the issues involved when selecting a sector size.

For instance, in the DVD, RSPC (Reed Solomon Product Code) coding requires 2064 Bytes whereas Picket code in Blu-Ray media requires 2074.5 Bytes [4].

In electronic memories (Flash), a sector size depends on the nature of the storage technology (single cell, double cell or multiple cells) and might be of 128, 256, 512...4096 Bytes.

A. Historical CHS address coding and sector size

Address coding is based on CHS symbols representing Cylinders, Heads and Sectors. Heads address upper and lower faces of platters, thus the number of heads should be even. In 24 bit CHS coding we have C= 10 bits, meaning that $2^{10}=1024$ is the maximum number of cylinders (tracks), H=8 bits means we have $2^8=256$ magnetic surfaces or 128 platters containing magnetic information whereas S=6 bits means that $2^6=64$ sectors per track are allowed.

The extended CHS addressing scheme moved over to 28 bits allocated to overcome the 13H BIOS address limitation problem with 16 bits allocated to C, 4 bits to H and 8 bits to S. This means we now have 65536 cylinders, 16 platter faces and 256 sectors per track.

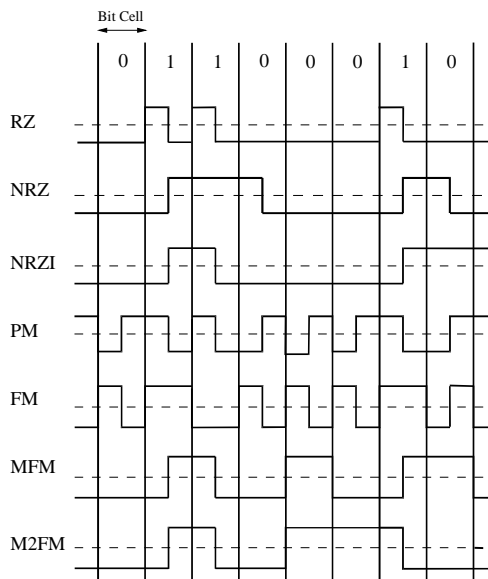


FIG. 4: Magnetic Line coding of information. Adapted from [5].

Historically, we display in table II and table III the characteristics of the 512-Byte Sector Format in the IBM XT and AT machines.

The main issue is that when errors occur randomly in an isolated fashion, they are easy to correct but when they occur in large groups (bursts), they get extremely hard to correct.

This is why codes such as Reed-Solomon ECC are very efficient in combating burst errors in several media ranging from magnetic to electronic and optic and many types of communication channels ranging from wireless mass transmission such as DAB (Digital Audio Broadcasting) and DVB (Digital Video Broadcasting) to wired over the

Name	Bytes	Description
Header	89	Gap, ID, Sync, Address Mark
Data	512	User data
Trailer	4	Error Correction Code (ECC)
Summary	605	Total Bytes/sector (84.6% efficient)

TABLE II: IBM XT 512-Byte Sector Format (see Mueller [3]).

Name	Bytes	Description
Header	71	Gap, ID, Sync, Address Mark
Data	512	User data
Trailer	4	Error Correction Code (ECC)
Summary	587	Total Bytes/sector (87.2% efficient)

TABLE III: IBM AT 512-Byte Sector Format (see Mueller [3]).

Internet Audio and Video streaming.

When disk size was about 100 MB, a 10^{-12} BER was largely sufficient since it implies reading the entire disk several hundred times before encountering a read error since $1/10^{12}$ is equivalent to 1/127 GB (see Table. II).

As density increased with time, Signal to Noise Ratio (SNR) started decreasing (like the super-paramagnetic effect) and consequently the number of parity bits increased from 4 to 50 Bytes (as in Table. IV).

Name	Bytes	Description
Header	15	Gap, Sync, Address Mark
Data	512	User data
Trailer	50	Error Correction Code (ECC)
Summary	577	Efficiency= [Sector size]/[Total size]= 512/577= 88.7%

TABLE IV: 512-Byte sector format in a HDD (see Mueller [3]).

The main issue is retrieving binary stored data that is reading it with least error.

This implies the use of EDC and ECC as displayed in fig.3.

B. Error Correction codes for magnetic recording

An important ingredient in binary signaling is the runlength i.e. the number of consecutive bits having the same value. The runlength concept gave birth to RLL (Runlength limited) coding used to perform line coding in information storage. RLL coding has a set of parameters, d, k, m, n , noted as $RLL(d, k, m, n)$ and shortened to $RLL(d, k)$. d is the minimal number of bits separating two consecutive ones whereas k is the maximum number of zeroes separating two consecutive ones. m is the minimum number of data bits to be encoded, n is the number of code bits for each of the m data bits. Thus m/n is the code rate. DR is the Density Ratio or data bits per flux reversal or $(d+1)(m/n)$, FR is frequency ratio given by maximum time over minimum time between transitions or $(k+1)/(d+1)$. Table V contains popular line codes in magnetic storage along with their RLL equivalence, whereas Fig. 4 shows the signal aspect of the RLL coding.

The evolution of ECC is required in order to follow density increase as displayed in Table. VII where single BER is correlated with disk size.

Digital magnetic recording is based on several peculiarities related to its intrinsic nature:

1. While Hamming codes are based on bits, magnetic recording ECC is more efficient when bits are grouped in Bytes or blocks (symbols) such as with Reed-Solomon coding.

Code Alternative names	Rate	Capacity	Efficiency	d, k	Density Ratio	Detection Window	DC Free
NRZ-L	1	1	100%	$0, \infty$	1	T	No
NRZ-M							
NRZI	1	1	100%	$0, \infty$	1	T	No
BiPolar Half	1	1	100%	$0, \infty$	1	T	No
BiPolar Full	1	1	100%	$0, \infty$	1	T	No
FM							
Biphase-M	1/2	0.6942	72%	0,1	0.5	T/2	Yes
Phase Encoding	1/2	0.6942	72%	0,1	0.5	T/2	Yes
FSK	1/2	0.6942	72%	0,1	0.5	T/2	Yes
Manchester							
BiPhase L	1/2	0.6942	72%	0,1	0.5	T/2	Yes
E-NRZ	7/8	1	87.5%	0,13	0.875	T	No
Miller							
MFM	1/2	0.5515	90.6%	1,3	1	T/2	No
DM	1/2	0.5515	90.6%	1,3	1	T/2	No
Miller	1/2	0.6509	76.8%	1,5	1	T/2	Yes
RLL 2,7	1/2	0.5174	96.6%	2,7	1.5	T/2	No

TABLE V: RLL coding. d is the minimal number of bits separating two consecutive ones whereas k is the maximum number of zeroes separating two consecutive ones (see Mueller [3]).

Code	Application	d	k	m	n	DR	FR
NRZ	Early HDD and magnetic tapes	0	∞	1	1	1	∞
NRZI	Early HDD and magnetic tapes	0	∞	1	1	1	∞
FM	Floppy disks	0	1	1	2	0.5	2.0
MFM	HDD, IBM 3330	1	3	1	2	1	2.0
Miller	HDD	1	3	1	2	1	2.0
Miller-Miller	HDD	1	5	1	2	1	3.0
M2FM	Floppy disks	1	4	1	2	1	2.5
RLL(1,7)	QIC Magnetic Tape	1	7	2	3	1.333	4.0
RLL(2,7)	HDD	2	7	2	4	1.5	2.67
Manchester	Magnetic Recording	0	1	1	2	0.5	2.0

TABLE VI: Modulation coding and RLL equivalence. Low density disks used FM, MFM and M2FM whereas high density uses RLL(1,7) and RLL(2,7). RLL(d, k, m, n) coding (shortened to RLL(d, k)), has parameters: d = Minimum number of consecutive zeroes allowed, k = Maximum number of consecutive zeroes allowed, m = Minimum number of data bits to be encoded, n = Number of code bits for each of the m data bits, DR is bit Density Ratio given by $(d + 1)(m/n)$, FR is the frequency ratio $FR = (k + 1)/(d + 1)$. Note that MFM is a variant of Miller. Miller-Miller coding was invented, in fact, by Mallinson and Miller [6]. QIC is Quarter-Inch Cartridge magnetic tape. Adapted from Ref. [5].

Bits read/Error	Tera-Bytes read/Error	Drive size (years)
10^{12}	0.125	5MB-100MB (1980)
10^{13}	1.25	100MB-100GB (1980-1990)
10^{14}	12.5	1GB-3TB+ (Since 1990)
10^{15}	125	RAID (Since 1990)

TABLE VII: Bits/Tera-Bytes (TB) Read per Error (Unrecoverable Read Error Rates) as a function of drive size. The initial 10^{-12} BER had to be decreased to about 10^{-15} in order to cope with current drive density and size (see Mueller [3]). RAID is Redundant Array of Identical Disks.

2. Errors increase with storage density, thus ECC must evolve with density
3. Errors occur in a burst mode involving large number of adjacent bits while it would be preferable they occurred in widely separated bits.
4. Coding gain is proportional to R^2 with R the rate code whereas in simple Additive White Gaussian Noise (AWGN) channels coding gain is proportional to R (see for instance Immink [6])
5. Partial response coding is quite efficient for pulse coding since it is based on correlative coding implying controlled ISI intentionally introduced in the signal in order to reduce ISI.

The read/write channel has tremendously evolved starting from simple peak-detection circuits and single burst-error correcting Fire codes, an extension of CRC codes, to more powerful coding techniques.

The introduction of maximum-likelihood detection [7] implied more accurate equalization and use of Viterbi detectors capable of accounting for signal-dependent noise and non-linear distortion [8]. Error-correction schemes quickly moved to using universally Reed-Solomon codes. HDD based on 512B sectors typically use ECC codes based on 10-bit symbols correcting 20 or more error symbols.

With HDD based on 4kB sectors, ECC codes based on 12-bit symbols are used. The next stage was the introduction of iterative detection [9] described by a typically gradual converging behavior as it ran iteratively between matching expected waveforms in the recording channel and respecting simultaneously the constraints imposed by parity checks.

ECC in hard drives is traditionally performed with CRC (Cyclic Redundancy Check [10]) or Fire codes [11, 12], their natural extension.

Later Reed-Solomon ECC was used in order to decrease BER down to 10^{-12} since CRC [10] could only achieve at that time a BER of about 10^{-10} . Reed-Solomon are constructed with Galois polynomials $GF(2^m)$ where m is the symbol size (see Table X). As an example, supposing we need to code a sequence of 577 Bytes (as in Table IV), $GF(2^9)$ is not adequate however $GF(2^{10})$ is.

C. Sector size increase issues

Increasing sector size offers, in fact, several benefits listed below:

1. Efficiency gain: Instead of making sequences of 8×512 B sectors with their overheads, one deals with a single sector of 4kB with a smaller size overhead as depicted in fig. 5.

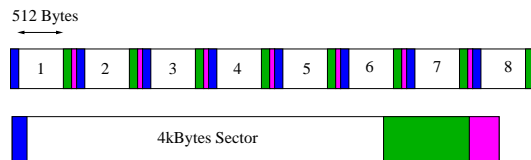


FIG. 5: (Color on line) Comparison of eight 512B and a single 4kB sector. Headers are in blue, ECC in Green and Trailer in Magenta. Headers comprise Gap, Sync and Address Mark (AM). Green blocks comprise CRC, ECC, Gap, Sync and AM. Trailer consists of a Gap. The 4kB sector is shorter than all eight 512B blocks while containing the same information. Moreover it is more efficient ECC wise and in combating burst errors.

2. Better efficiency of ECC: 100 Bytes of ECC for a single 4kB block are as reliable as 320 Bytes of ECC for eight 512 Byte blocks. From table X we infer 9.8% coding overhead in the 512B sector and a 2.4% in the 4K case.
3. Better handling of burst errors. Ordinary ECC can correct up to 50 Byte bursts. When burst error size goes from 50B to 200B, 512B sectors are strongly affected implying that increasing sector size will be more beneficial in the case of large burst errors.

In table VIII, are displayed the 4k-Byte sector characteristics in a HDD and in table IX, several sector sizes with their respective performance are compared.

Name	Bytes	Description
Header	15	Gap, Sync, Address Mark
Data	4096	User data
Trailer	100	Error Correction Code (ECC)
Summary	4211	Total Bytes/sector (2.4% efficient)

TABLE VIII: 4k-Byte Sector Format in a HDD (see Mueller [3]).

Sector size	512B	1kB	2kB	4kB
Overhead:				
ECC (Bytes)	64	94	140	206
Sync (Bytes)	72	72	72	72
Data cell (Bytes)	674	1216	2286	4400
Efficiency:				
ECC	0.889	0.916	0.936	0.960
Data (timing)	0.855	0.919	0.957	0.978
Zone	0.917	0.966	0.978	0.978
Servo	0.843	0.843	0.843	0.843
Split	1.000	0.999	0.999	0.999
Total track	0.587	0.685	0.737	0.764
Relative gain	100%	116%	125%	130%

TABLE IX: Effect of sector size on capacity gain for 100 Gb/in² media (84mm disk @ 15,000 rpm) as compared to 512B case. Adapted from reference [13].

V. ELECTRONIC STORAGE

Electronic Flash technology has been developing at a remarkable pace as displayed in fig. 6.

Flash technology is important for mobile electronic devices (e.g. digital camera, smartphone, portable SSD drives, key drives, tablets and notebooks).

The Flash transistor is a MOSFET transistor containing within its Oxide layer a thin metallic film called floating gate (hence the name Flash) than can trap electrons by application of a positive voltage to the gate.

At the beginning of Flash technology a single bit corresponded to a single Flash transistor (SLC or single level cell). Later development led to 2-bit/transistor (MLC or Multiple level cell), then 3-bit/transistor (TLC or triple level cell) and finally 4-bit/transistor (QLC or quadruple level cell).

NAND Flash memory is the most aggressively scaled technology among electronic devices because the NAND Flash memory cell have relatively simple structure compared with others and there is severe industrial competition. Also, there were many technological innovations (such as multi-level cell and double patterning lithography) that resulted in cell size shrinking faster than dictated by Moore's law.

Sector size	Galois field	ECC Size	Raw BER	BER after ECC
512B	GF(2 ¹⁰)	50B	10 ⁻⁵ - 10 ⁻⁶	10 ⁻¹¹ - 10 ⁻¹²
4kB	GF(2 ¹²)	100B	10 ⁻³ - 10 ⁻⁴	10 ⁻¹¹ - 10 ⁻¹²

TABLE X: Reed-Solomon ECC characteristics for 512B and 4kB sectors. Galois fields such as GF(2^m) are polynomials where m is an integer equal to the symbol size. The number of elements in GF(q^m) is q^m . Sector 512B requires GF(2¹⁰) whereas 4kB requires GF(2¹²) [14].

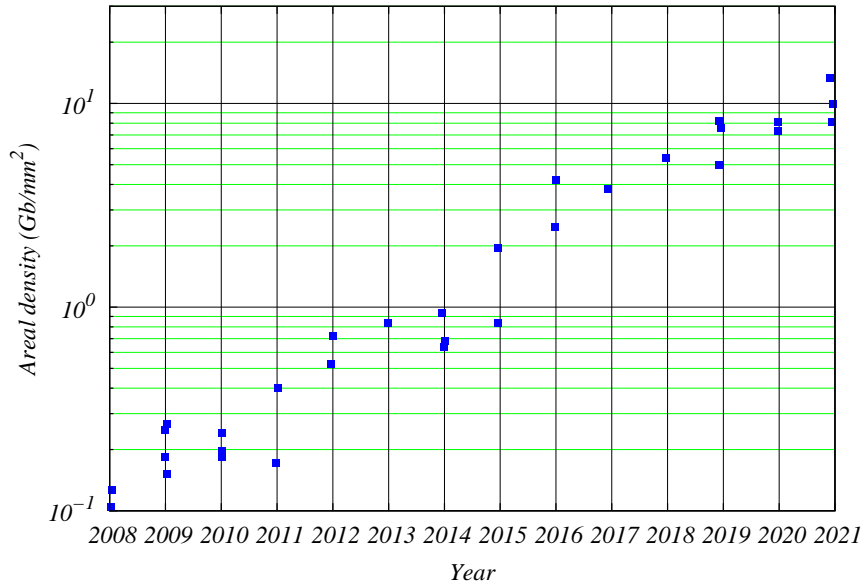


FIG. 6: (Color on line) Evolution of Flash areal density over the period 2008-2021. The data contain 2D and 3D MLC, TLC and QLC Flash types. The transition from 2D to 3D started circa year 2014.

A. Error Correction codes for NAND Flash

The number of parity Bytes depends on the ECC type [15] with its performance hierarchy discussed below (see Fig.7):

- **Hamming Codes:**
A block Hamming code can correct a single bit error and detect occurrence of multiple bit errors. It is also able of detecting most double bit errors.
- **Reed-Solomon Codes:**
Reed-Solomon codes are used in communications and storage devices to handle burst errors. When Flash devices (MLC) moved to 2kB or 4kB pages, ECC required 4 bit corrections.
- **Bose-Chaudhuri-Hocquenghem (BCH) Codes:**
BCH codes are capable of handling large random errors whereas Reed-Solomon codes are used for correcting burst errors. A $BCH(n, k; t)$ code over Galois Field $GF(2^m)$ (see Appendix) with n bit codeword length and k bit message length can correct up to t error bits.
- **Low Density Parity Check (LDPC) Codes:**
The advent of TLC and QLC (3 and 4 level cell) along with scaling down to provide further Flash storage implies BCH correction to be replaced progressively by LDPC codes as the most appropriate ECC or a concatenation of BCH and LDPC, as in magnetic recording.

Let us compare parity sizes for Hamming, Reed-Solomon, and BCH implementations:

- **Hamming:**
Block Hamming codes correct only a single error and require $2 \log_2(n)$ of parity for a data block with n data bits.
A 512B data block consists of 4096 (2^{12}) bits, thus a Hamming code requires $2 \log_2(n)$ with $n = 12$ or 24 parity bits.
- **Reed-Solomon:**
Parity required for an RS code depends on the symbol size, Galois field size, and ECC level provided by the code. This means that if we have symbols of size m , the Galois field $GF(2^m)$ size is m , chosen generally to be the minimal value corresponding to a given selected block size. Thus the number of parity bits is $2 \times m \times \text{ECC}$.

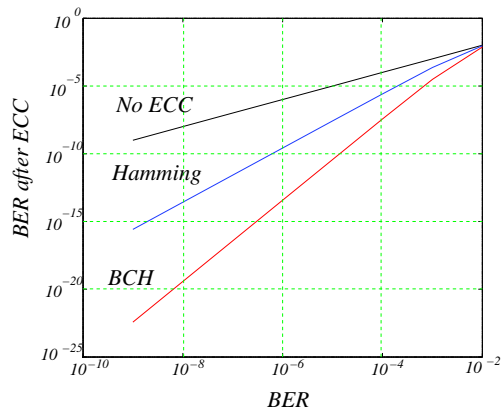


FIG. 7: Effect of coding on BER in going from raw to Hamming to BCH ECC illustrating performance hierarchy of ECC. Adapted from ref. [16].

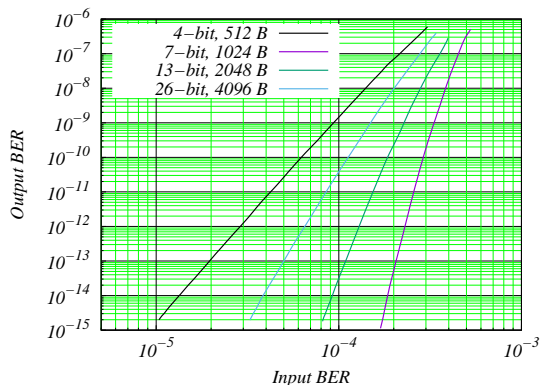


FIG. 8: ECC number of bits and sector size effects on corrected BER in Flash devices. Output BER improvement is about 10 orders of magnitude with respect to input BER after BCH based ECC coding. Adapted from Ref. [17].

For example, an RS over a 512B data block consisting of 9 bit symbols capable of supporting 8 symbol correction would require $2 \times 9 \times 8$, or 144 bits of parity.

- BCH:

Parity required for BCH is dependent on the Galois field $GF(2^m)$ size m (determined by the data block size) and the ECC correction level. The number of parity bits can be computed as $m \times ECC$.

The following table XI shows the number of ECC bits/Bytes required per correction block.

Block size	ECC Level	ECC Number of bits
512B	ECC 8	$13 \times 8 = 104$
512B	ECC 16	$13 \times 16 = 20826$
1024B	ECC 24	$14 \times 24 = 33642$
1024B	ECC 40	$14 \times 40 = 56070$

TABLE XI: Number of ECC bits/Bytes required per correction block.

VI. OPTICAL STORAGE

Optical storage spans CD, DVD and Blu-Ray disks as well as Holographic Video Disks (HVD) that has not been widely developed for technical reasons.

In contrast to magnetic addressing based on 1D tracks (cylinders) and heads (platter face) such as in HDD, optical storage is based on a single spiral track with a linear addressing mode. Holographic is more complicated since it is based on 2D coding and addressing.

A. Error Correction codes for Optical storage

Optical storage is particularly prone to burst errors (resulting from surface scratches for example) and Reed-Solomon ECC techniques equivalent to the RLL family are the main choice. EFM (Eight to Fourteen Modulation, see Table XII) is used for CDROM and EFM+ is used for DVD whereas 17PP (17 originating from RLL(1,7) coding and PP being Parity Preserving) codes are used in Blu-Ray discs (see Table XII).

Optical storage characteristics and ECC are shown in Table XII.

Media	λ	NA	Capacity	RLL code	ECC code
CDROM	780 nm	0.45	700 MB	EFM	CIRC
DVD	680 nm	0.55	4.7 GB	EFM+	RSPC
Blu-Ray	405 nm	0.85	25 GB	17PP	Picket code

TABLE XII: Optical media characteristics. EFM is based on RLL(2,10) transforming 8 bit symbols into 14 bit symbols and linking them with 3 extra bits whereas EFM+ is based on RLL(2,10) transforming 8 bit symbols into 14 bit symbols and linking them with 2 extra bits. 17PP is RLL(1,7) with parity preservation. EFM should be compared to Eight to Ten Modulation used in magnetic DAT (Digital Audio Tape) storage. CIRC or Cross-interleaved Reed-Solomon coding uses two Reed-Solomon codes with an interleave between. RSPC or Reed-Solomon Product Code consists of arranging data in a matrix and detecting parities along the horizontal and vertical directions.

	DVD	Blu-ray
Sector size	2064B	2074.5B
ECC block	32kB	64kB
ECC code rate	0.866	0.852
ECC nature	Product code	Code & Subcodes*
Number of corrected burst errors (200B ECC)	5-14	32-49 (349 μm)
Number of corrected burst errors (600B ECC)	3-4	10-16 (1047 μm)
Max correctable burst length (ECC)	2912B	9920B

TABLE XIII: Burst error correction in DVD (RSPC) and Blu-Ray (Picket code) discs. *In the Blu-Ray subcodes cover long distance & burst indicators.

APPENDIX A: DETAILED DESCRIPTION OF LINE AND RLL CODES

Line coding specifies how a string of binary digits are converted to a line code waveform. There are two major classes of binary line codes: level codes and transition codes. Level codes carry information in their voltage level, which may be high or low for a full bit period or part of the bit period. Level codes are usually instantaneous since they typically encode a binary digit into a distinct waveform, independent of any past binary data. However, some level codes do exhibit memory. Transition codes carry information in the change in level appearing in the line code

waveform. Transition codes may be instantaneous, but they generally have memory, using past binary data to dictate the present waveform. There are two common forms of level line codes: one is called return to zero (RZ) and the other is called non-return to zero (NRZ).

Line coding formats are further classified according to the polarity of the voltage levels used to represent the data. If only one polarity of voltage level is used, i.e., positive or negative (in addition to the zero level) then it is called unipolar signaling. If both positive and negative voltage levels are being used, with or without a zero voltage level, then it is called polar signaling.

- RZ coding
The level of the pulse returns to zero for a portion of the bit interval.
- NRZ coding
The level of the pulse is maintained during the entire bit interval.
- Unipolar RZ
A binary 0 is a zero voltage level during the entire bit period. A binary 1 is a non-zero voltage level during a portion of the bit period, usually for half of the bit period, and a zero voltage level for rest of the bit period.
- Polar NRZ
A binary 1 is a positive voltage and a binary 0 is a negative voltage over the full bit period.
- Polar RZ [Bipolar, Alternate Mark Inversion (AMI)]
A binary 1 is represented by alternating the positive and negative voltage levels, which return to zero for a portion of the bit period, usually half the bit period. A binary 0 is represented by a zero voltage level during the entire bit period. Alternate mark inversion (AMI) means that ones (marks) are represented by alternating positive and negative pulses.
- Manchester Coding (Split Phase or Digital Biphas)
- Phase Modulation (PM)
- Frequency Modulation (FM)
- Delay Modulation (Miller Code) or Modified Frequency Modulation (MFM)
- Modified MFM (M2FM)

APPENDIX B: GALOIS, CRC AND PARTIAL RESPONSE POLYNOMIALS

Testing a communication channel requires generation of binary sequences in a fast and controllable way, akin to Random Number Generation in Monte-Carlo simulation. The main goal of PRBS (Pseudo Random Binary Sequence) generation, is to draw 0 or 1 in an equally probable fashion in order to have highly efficient crypting of communication as in spread-spectrum communications [2] used in wireless telephony or any message over some telecommunication channel. PRBS are useful for testing communication lines, storage media, coding techniques... A particularly efficient method for producing PRBS is based on primitive (akin to prime number) polynomials modulo

2 or Galois polynomials [18] called Galois Fields GF(2) with the following arithmetic:

$$0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0.$$

\oplus is the usual symbol for modulo 2 arithmetic corresponding to the logical XOR operation.

The coefficients of primitive polynomials modulo 2 are zero or one e.g. $x^3 + x^2 + 1$. They cannot be decomposed into a product of simpler modulo 2 polynomials (akin prime numbers). Hence $x^2 + 1 = (x + i)(x - i)$ with $i = \sqrt{-1}$ cannot be decomposed into simpler polynomials with real coefficients. When this polynomial is viewed as a modulo 2 Galois polynomial, it is not primitive since it can be decomposed into a product of simpler polynomials $x^2 + 1 \equiv x^2 + 2x + 1 = (x + 1)(x + 1)$ since in modulo 2 arithmetic the term $2x$ is equivalent to 0 according to the rule $(1 \oplus 1 = 0)$.

Galois fields such as GF(q^m) are polynomials where q is a prime number and m is an integer equal to the symbol size. The number of elements in GF(q^m) is q^m . For instance if we are dealing with simple bits as in PRBS, we use GF(2) ($q = 2, m = 1$) but if we are using Bytes, then we use GF(2^3) ($q = 2, m = 3$).

Shift Register taps	Galois Polynomial
(1, 0)	$x + 1$
(2, 1, 0)	$x^2 + x + 1$
(3, 1, 0)	$x^3 + x + 1$
(4, 1, 0)	$x^4 + x + 1$
(5, 2, 0)	$x^5 + x^2 + 1$
(6, 1, 0)	$x^6 + x + 1$
(7, 1, 0)	$x^7 + x + 1$
(8, 4, 3, 2, 0)	$x^8 + x^4 + x^3 + x^2 + 1$
(9, 4, 0)	$x^9 + x^4 + 1$
(10, 3, 0)	$x^{10} + x^3 + 1$

TABLE XIV: List of the first 10 Galois polynomials. They are generated with Shift Registers with connection nodes corresponding to the polynomial powers [18]. The + operation is performed with an XOR operation.

The method for producing PRBS requires only a single shift register n bits long and a few XOR or modulo 2 bit addition operations (\oplus gates).

There is one term for each non-zero coefficient in the primitive polynomial except the constant (zero bit) term. Table XIV contains a list of polynomials for $n \leq 10$, showing that for a primitive polynomial of degree n , the first and last term are 1.

CRC (Cyclic Redundancy Check) polynomials allow to check the integrity of a file or a message transmitted over some communication channel. They are based on special polynomials called generators as displayed in Table XV.

CRC coding is based on considering the message as a polynomial $M(x)$ where the i -th bit of the message multiplies x^i . The least significant bit of the message is numbered 0. $G(x)$ is the generator polynomial (divisor) of degree r . The polynomial division takes place in Galois Field GF(2), meaning that all arithmetic operations are performed modulo 2.

The check sum value $C(x)$ is calculated by division of $x^r M(x)$ by $G(x)$ i.e. $x^r M(x) = Q(x)G(x) + C(x)$, with $Q(x)$ the quotient and $C(x)$ the remainder. The x^r multiplication is equivalent to r bit shifts akin to multiplication by 10^r in the decimal system.

Subtracting this remainder from the dividend $x^r M(x)$ yields a polynomial $T(x) = x^r M(x) - C(x)$, which is exactly divisible by $G(x)$. $T(x)$ is then transmitted through some communication channel and received later as an eventually corrupted $R(x)$. A non-zero remainder resulting from division of $R(x)$ by $G(x)$ indicates corruption of the message.

Partial response coding was initially developed to combat ISI [2] when signaling was based on sinc functions. It is a signal shaping method meant to reduce ISI in communication systems and recording by increase of storage density in magnetic systems.

The notion of Hamming distance [1] is very valuable in order to estimate their potential ISI. For instance if we have two 3-bit symbols (000) and (111) belonging to a 3D symbol space, their Hamming distance is the XOR addition $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$, applied separately to every bit, yielding 111 and afterward adding the bits

Code Name	$G(x)$	Application
CRC-1	$x + 1$	Parity Check
CRC-4-ITU	$x^4 + x + 1$	ITU G.704
CRC-5-ITU	$x^5 + x^4 + x^2 + 1$	ITU G.704
CRC-5-USB	$x^5 + x^2 + 1$	USB
CRC-6-ITU	$x^6 + x + 1$	ITU G.704
CRC-8-ATM	$x^8 + x^2 + x + 1$	ATM
CRC-8-CCITT	$x^8 + x^7 + x^3 + x^2 + 1$	1-Wire bus
CRC-8-Maxim	$x^8 + x^5 + x^4 + 1$	1-Wire bus

TABLE XV: Some CRC polynomials and their applications. ITU is International Telecom Union. USB is Universal Serial Bus. ATM is Asymmetric Transfer Mode based on Statistical Multiplexing of digital data.

in a decimal fashion to get finally 3.

Coding gain is given by [1] $G = 10 \log_{10} \left[\left(\frac{d_{min}}{d_{min,u}} \right)^2 \left(\frac{E_s}{E_{s,u}} \right) \right]$ where Euclidean distances $d_{min}, d_{min,u}$ are considered between signal constellation symbols in the coded and uncoded cases (noted u). $E_s, E_{s,u}$ are average signal energies for the coded and uncoded cases.

Coding is considered successful when it increases the distance (Hamming or Euclidean) between constellation symbols. Several examples of coding distances are displayed in Table XVI to illustrate ISI combating with partial response coding as reflected in distances (Minimal and Distance to Matched Filter [1]) between constellation symbols.

Partial Response (PR) coding is based on Delay polynomials (see Table XVI) where a Huffman time delay operator D acts on bit signals a_k at instant k to produce $Da_k = a_{k-1}$. In fact, digital data patterns are given by sequences of bits a_k and when a magnetic head reads the signal by responding to magnetization changes, its action is equivalent to a differentiation. Head action can be expressed mathematically as $a_k - a_{k-1} = a_k - Da_k = (1 - D)a_k$. The operator given by $(1 - D)$ will result in +1 or -1 responses, depending on the direction of magnetization change. Table XVI gives several examples of PR coding along with the corresponding Delay polynomials and coding gains resulting from distance increase between symbols reducing ISI.

Code Name	Delay Polynomial	d_{min}^2	d_{MF}^2	d_{min}/d_{MF} (dB)
PR1	$(1 + D)$	2	2	0
PR2	$(1 + D)^2$	4	6	-1.76
EPR2	$(1 + D)^3$	10	20	-3.01
Dicode	$(1 - D)$	2	2	0
PR4	$(1 - D)(1 + D)$	2	2	0
EPR4	$(1 - D)(1 + D)^2$	4	4	0
E2PR4	$(1 - D)(1 + D)^3$	6	10	-2.22
E3PR4	$(1 - D)(1 + D)^4$	12	26	-3.36

TABLE XVI: Minimal and Distance to Matched Filter (MF) in Partial Response Channels [1] yielding coding gains. EPR means Extended Partial Response.

Table XVII gives classes of PR coding transfer function $H(z)$ along with the corresponding Delay polynomials and coding gains resulting from distance increase between symbols reducing ISI. Coding entails transforming a bit sequence $\{a_k\}$ into another one $\{b_k\}$ such that [19] $A(D) = \sum_k a_k D^k$, is related to $B(D) = \sum_k b_k D^k$ with a Delay polynomial $P_n(D) = \sum_{k=0}^{n+1} p_k D^k$, such that an output-input relation is expressed as: $B(D) = P_n(D)A(D)$. Thus $P_n(D)$ is an impulse response that could be Z -transformed [19] into a transfer function $H(z)$ as displayed in Table XVII. $H(z)$ is a $1/z$ polynomial [2] type transfer function since $1/z$ is the Z -transform of the delay operator D .

Class	Delay Polynomial	Transfer function $H(z)$	Comments
I	$(1 + D)$	$1 + z^{-1}$	Corresponds to PR1 (Table XVI), Null at Nyquist
II	$(1 + D)^2$	$1 + 2z^{-1} + z^{-2}$	Corresponds to PR2 (Table XVI), Two nulls at Nyquist
III	$(1 + D)(2 - D)$	$2 + z^{-1} - z^{-2}$	$(2 - D)$ factor is unusual
IV	$(1 + D)(1 - D)$	$1 - z^{-2}$	Corresponds to PR4 (Table XVI), Null at DC and at Nyquist
V	$-(1 + D)^2(1 - D)^2$	$-1 + 2z^{-2} - z^{-4}$	Null at DC

TABLE XVII: Classes of Partial-response signaling characteristics with nulls at DC or Nyquist frequency. Adapted from J.B Anderson [20] chapter 3 on Carrier Transmission. Class I is also called Duo-binary [2] signaling.

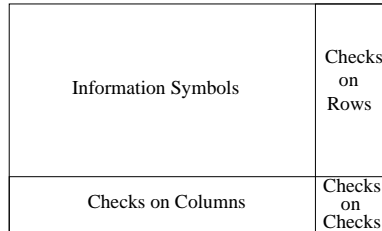


FIG. 9: A product code like RSPC (Reed Solomon Product Code) consists of arranging data in 2D arrays and detecting parities along the horizontal and vertical directions. RSPC coding in DVD requires 2064 Bytes, whereas Picket code in Blu-Ray media requires 2074.5 Bytes. Picket code ECC is tailored particularly for handling burst errors in Blu-Ray Optical media.

APPENDIX C: REED-SOLOMON CODES

Reed-Solomon codes are linear and cyclic. They handle bits as well as symbols. They are good in correcting burst errors. They are based on Galois fields such as $GF(q^m)$ where q is a prime number and m is an integer equal to the symbol size.

When we use 512 Byte blocks we have an overhead of 65 Bytes (15 for gap,sync,address and 50 for ECC). Thus we need to code 577 Bytes (see Table IV) and $GF(2^9)$ is not sufficient, given that $2^9 = 512$, consequently we use $GF(2^{10})$.

In the 4k sector case we have 15 (gap,sync,address)+100 (ECC) Bytes, thus we use $GF(2^{13})$ and not $GF(2^{12})$.

Interestingly, Reed-Solomon codes can be arranged along several directions as for instance the 2D product code RSPC (Reed Solomon Product Code) used in DVD where parities are detected along a vertical as well as a horizontal direction (see Fig. 9).

-
- [1] J. L. LoCicero and B. P. Patel, Chapter 6 in *Mobile Communications Handbook*, S. S. Suthersan (Ed.), Boca Raton: CRC Press (1999).
 - [2] A. B. Carlson and P. B. Crilly, *Communication systems: An Introduction to Signals and Noise in Electrical Communication*, 5th Edition, McGraw-Hill, New York (2010).
 - [3] S. Mueller, *Repairing and Upgrading PC's*, 20th Edition, Pearson Education (2012).
 - [4] X. Liu, HB Jia and C. Ma, *Error correction codes for Optical Disc Storage*, in Proceedings of SPIE, Vol. 5643, SPIE Bellingham, WA (2005).
 - [5] J. Cavanagh, *Magnetic Recording Fundamentals*, Chapter 7 in *Digital Design and Verilog HDL Fundamentals*, 1st Edition, Boca Raton: CRC Press (2008).
 - [6] K.E.S Immink, IEEE Trans. Comm., **37**, 413 (1989), K.E.S Immink, *Codes for Mass Data Storage Systems*, Second Edition, Shannon Foundation Publishers, Eindhoven, The Netherlands (2004).
 - [7] R. Wood, S. Ahlgrim, K. Hallamasek, R. Stenerson, IEEE Trans. Magn. MAG-20 (5), 698 (1984).
 - [8] E. Eleftheriou, *Signal processing for magnetic-recording channels*, in: *Encyclopedia of Telecommunications*, J. Proakis (Ed.), Vol. 4, Wiley, New York, 2247 (2003).
 - [9] Z. Wu, *Coding and Iterative Detection for Magnetic Recording Channels*, Kluwer Academic Publishers, (2000).

- [10] D.E. Knuth, *The Art of Computer Programming*, in Section 4.6, Volume 2: *Semi-numerical Algorithms*, Third Edition, Addison-Wesley (1997).
- [11] R. Wood, *Journal of Magnetism and Magnetic Materials* **321**, 555 (2009).
- [12] E. Fujiwara, M. Yashiro and T. Furuya, in *Essentials of Error-Control Coding techniques*, Academic Press (1990).
- [13] D. Cheng, M. Hassner, P. Hodges, B. Lamberts and R. Wood, *NSIC Discussion, IBM Corporation*, (1998).
- [14] H. Matsui and S. Mita, *A New Encoding and Decoding System of Reed-Solomon Codes for HDD*, *IEEE Trans. Mag.*, **45**, 3757 (2009).
- [15] E. Deal, *Hamming, RS, BCH, LDPC Alphabet Soup of NAND ECC Parity Bytes*, *Cyclic Design* (2011).
- [16] C-C Chu, Y-M Lin, C-H Yang, and H-Ch Chang *A Fully Parallel BCH Codec with Double Error Correcting Capability for NOR Flash Applications*, 1605, *IEEE, ICASSP* (2012).
- [17] N. Duann, *Flash Summit Meeting, Santa-Clara, USA*, Silicon Motion Inc. (2009).
- [18] W. H. Press, W. T. Vetterling, S. A. Teukolsky and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing* Third Edition, Cambridge University Press, New-York (2007).
- [19] H. K. Thapar and A. M. Patel, *A Class of Partial Response Systems for increasing storage density in magnetic recording* *IEEE Trans. Mag.* **23**, 3666 (1987).
- [20] J. B. Anderson, *Digital Transmission Engineering*, Chapter 3, Second Edition, IEEE Press and Wiley-Interscience (2005).